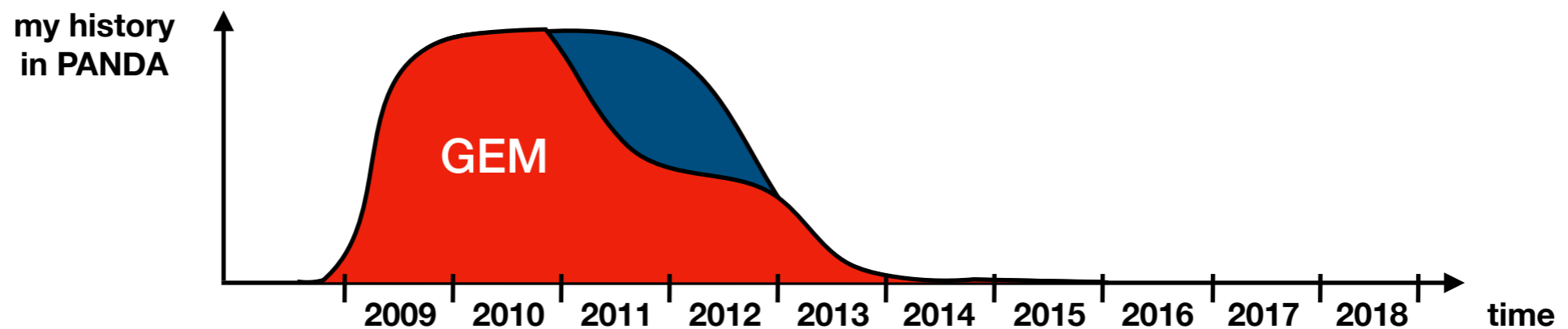


BarrelTrackFinder

Radoslaw Karabowicz, GSI

history

- GEM Tracker software development for Panda from 2009 till 2012:



- No global tracking back then, only STT track finder by Gianluigi Boca with extensions to MVD and GEM;
- BarrelTrackFinder developed and presented in 2010.

idea

- Simple;
- No detector prioritization: treat MVD, STT, GEM hits the same, put the hits in one array, and randomly take one hit after another;
- Reduce combinatorics: do not try to compare each and every hit, get rid of triple hit loops;

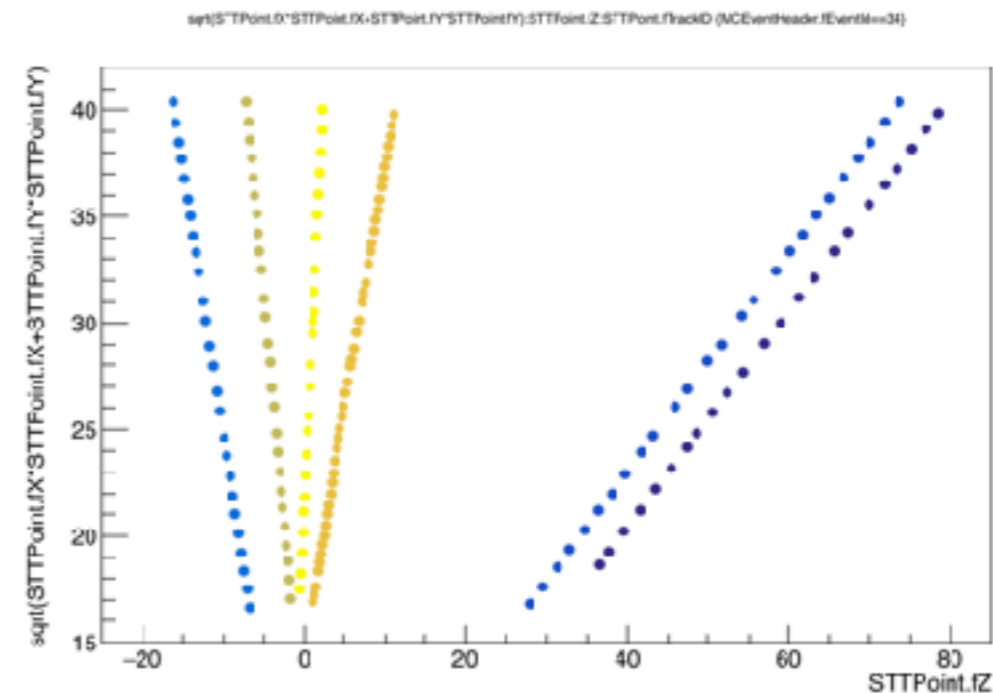
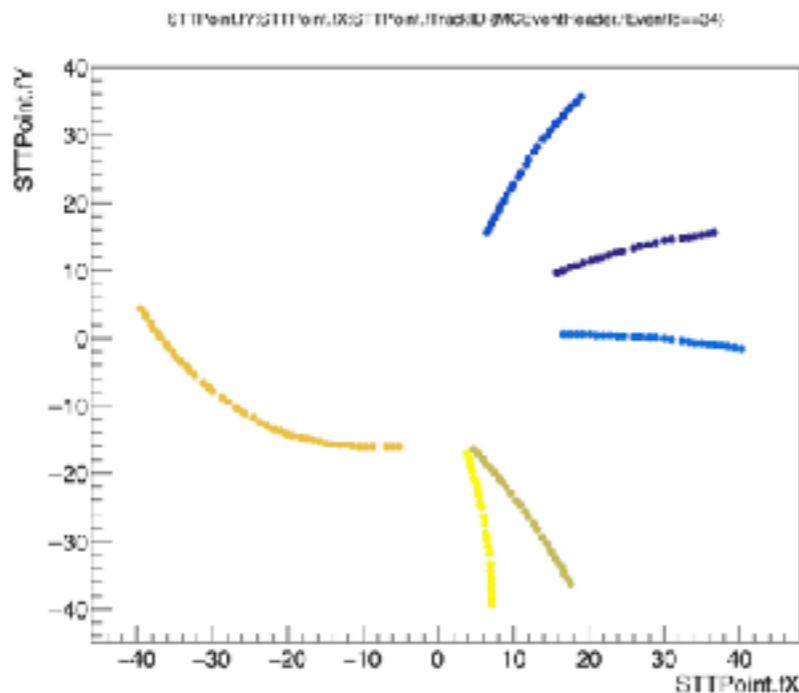
algorithm

```
std::vector<Track> possibleTracks(0);
std::vector<Int_t> unusedHits(0);
for ( Int_t ihit = 0 ; ihit < nofAllHitsInEvent ; ihit++ ) {
    Int_t matching = 0;
    for ( Int_t itrack = 0 ; itrack < possibleTracks.size() ; itrack++ )
        matching += MatchHitToTrack(ihit,itrack);
    if ( matching > 0 ) continue;
    matching = 0;
    for ( Int_t iunh = 0 ; iunh < unusedHits.size() ; iunh++ )
        matching += MatchHitToHit(ihit,iunh);
    if ( matching > 0 ) continue;
    unusedHits.push_back(ihit);
}

CleanTracks();
```

Track

1. emitted from the nominal vertex (0,0,0);
2. simple helix representation:
 - circle on the X-Y plane (X_c, Y_c, R_c);
 - straight line on the R-Z plane ($\Delta z / \Delta \varphi$)



MatchHitToTrack


MVD
GEM

STT

MatchHitWithZInfoTT

MatchSkewedSttHitTT

MatchParallelSttHitTT

distance
hit XY - 
track circle
on XY plane


(if tracks' $\Delta z/\Delta\varphi$ known)

+

difference of
track's $\Delta z/\Delta\varphi$
and
hit's $\Delta z/\Delta\varphi$



check if
skewed hit
pierces the
cylinder
described by the helix

distance
hit XY - 
track circle
on XY plane

Hit is matched to track if certain criteria are satisfied

MatchHitToHit

MVD
GEM

STT

Disregard

All different possibilities (1, 2 or 4)
are stored in one track

(0,0,0)

CleanTracks

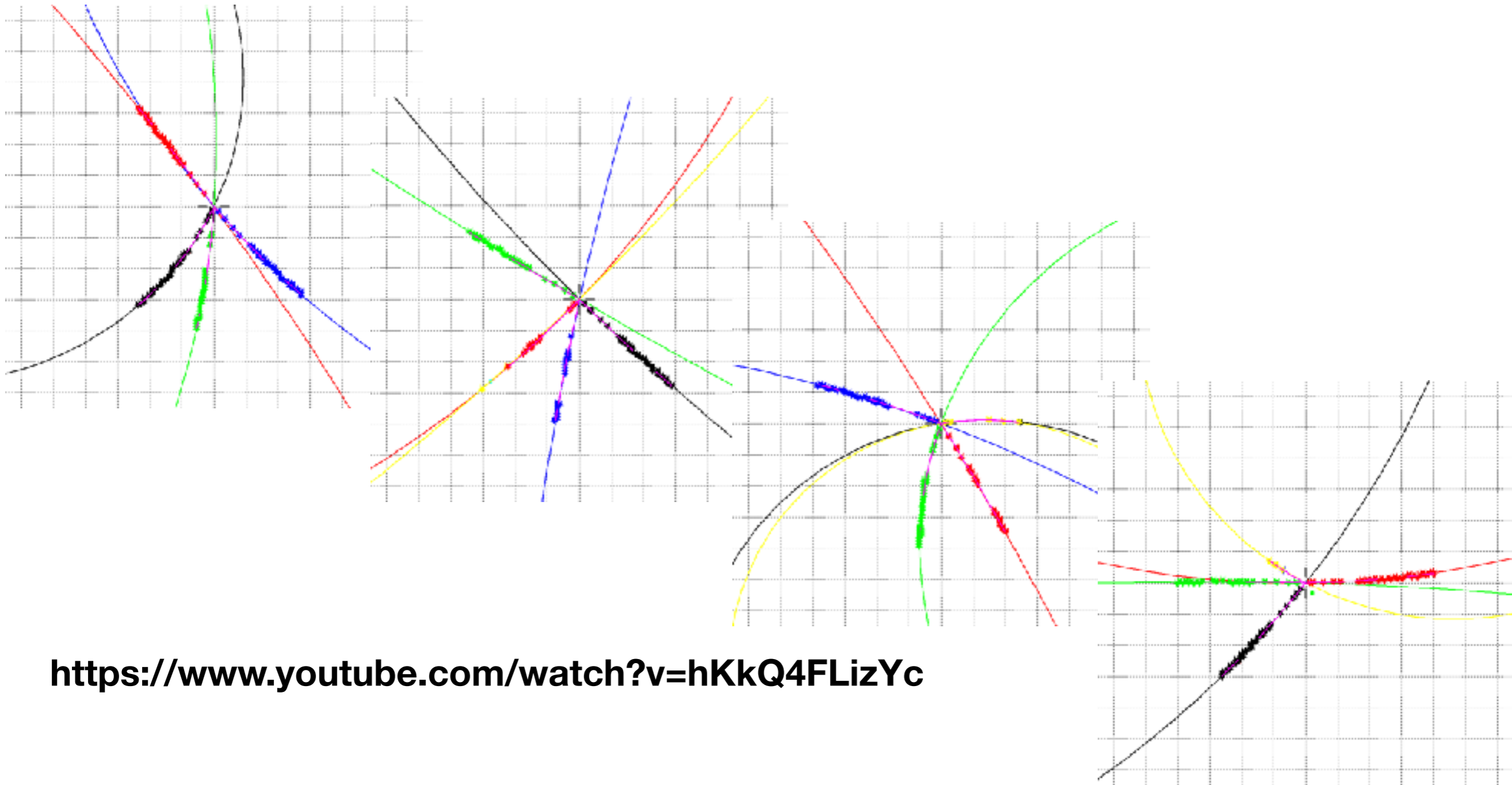
- Remove short tracks;
- Reuse unused hits;
- Merge tracks with similar parameters and same hits;

algorithm

```
std::vector<Track> possibleTracks(0);
std::vector<Int_t> unusedHits(0);
for ( Int_t ihit = 0 ; ihit < nofAllHitsInEvent ; ihit++ ) {
    Int_t matching = 0;
    for ( Int_t itrack = 0 ; itrack < possibleTracks.size() ; itrack++ )
        matching += MatchHitToTrack(ihit,itrack);
    if ( matching > 0 ) continue;
    matching = 0;
    for ( Int_t iunh = 0 ; iunh < unusedHits.size() ; iunh++ )
        matching += MatchHitToHit(ihit,iunh);
    if ( matching > 0 ) continue;
    unusedHits.push_back(ihit);
}

CleanTracks();
```

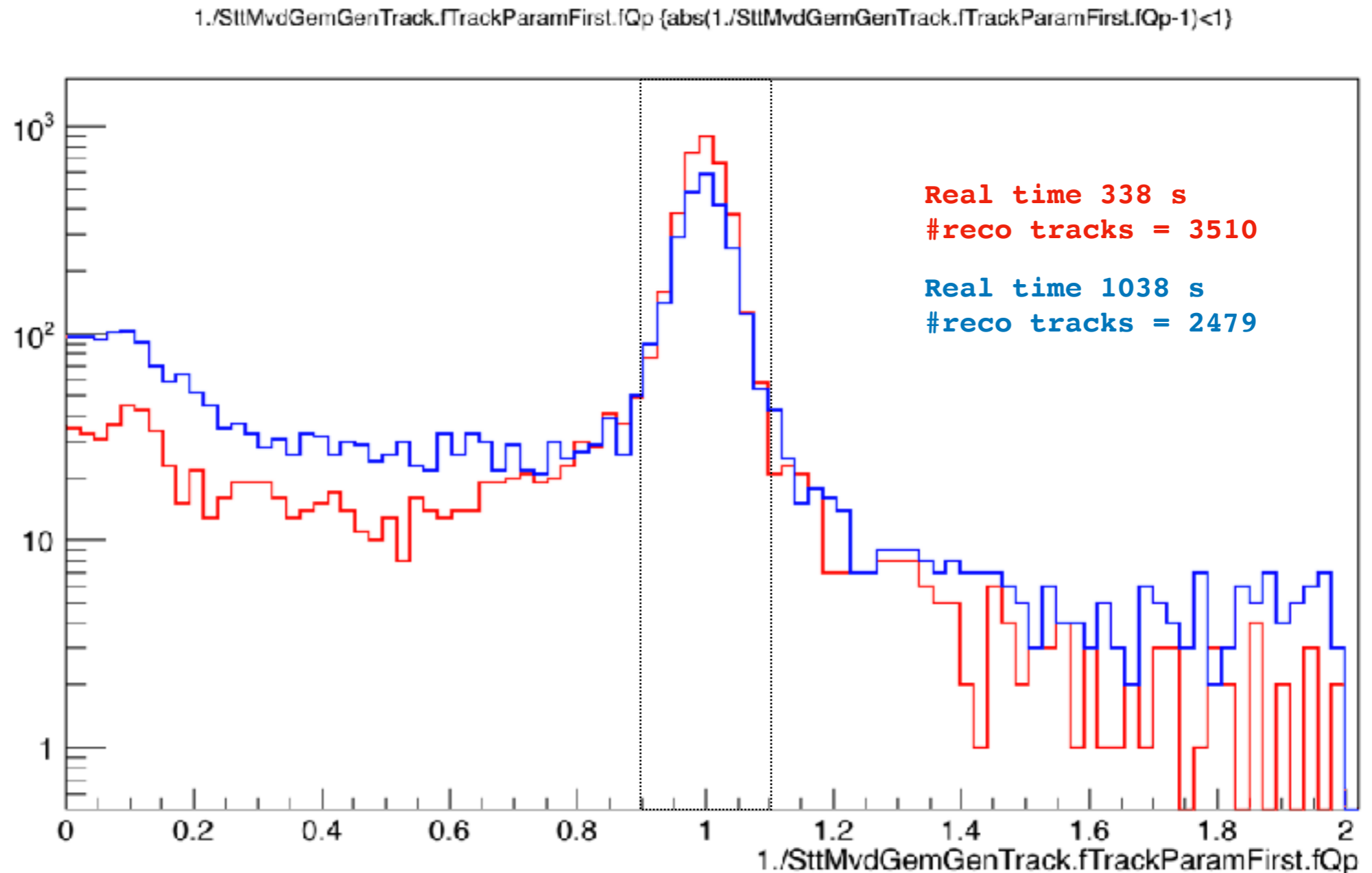
live demo



<https://www.youtube.com/watch?v=hKkQ4FLizYc>

PndSttMvdGemTracking vs PndBarrelTrackFinder

1000 events:
5 pions
 $p=1$ GeV/c
 $10^\circ < \theta < 120^\circ$
 $0^\circ < \varphi < 360^\circ$



final remarks

- code developed and tested 8 years ago;
- some work is needed to bring it back to life;
- much more work is needed to adapt the code to the time-based reconstruction workflow;
- the code is available to test and develop under:
`pandaroot/trunk/tracking/PndBarrelTrackFinder.{h,cxx}`